Lab 7: Distributed Arithmetic

This laboratory will introduce you to the Distributed Arithmetic method of filter implementation. This is a very area-efficient design technique for implementing filters, transforms, and more on FPGAs.

Task 1: Designing a Serial Distributed Arithmetic FIR Filter

Using the same coefficients from Laboratory 6, implement a Distributed Arithmetic FIR filter. Use the Serial Distributed Arithmetic (SDA) form by implementing a single distributed arithmetic look-up table (DALUT) as shown in Figure L7.1.
Compile your project and download it to the Digilab-IIE board. Verify it’s functionality by using MATLAB (and the MATLAB-Xilinx interface introduced in Laboratory 4) to generate ideal and experimental frequency response magnitude plots and overlay them on the same figure. Finally, compare the size of this implementation with the various implementations from Laboratory 6.

**Task 2: Designing an IIR Filter**

Use MATLAB’s Filter Design and Analysis Toolbox, or any other method you choose, to design an IIR filter with the following specifications.

- Stopband attenuation must be at least 40dB
- Passband ripple (if any) should not be more than 1dB
- The filter order should not be higher than 12
- passband $300\,\text{Hz} < f < 3300\,\text{Hz}$
- stopbands $0 \leq f < 200\,\text{Hz}$, $4000 < f \leq f_s/2$

Note, it is important to remember that when designing a filter in MATLAB the specified order is for the lowpass prototype and not for the bandpass filter. The bandpass filter actually has an order twice that specified.

**Task 3: Designing an Serial Distributed Arithmetic IIR Filter**

Using the coefficients you generated in Task 2, implement a Distributed Arithmetic IIR filter. Use the Serial Distributed Arithmetic (SDA) form by implementing a single DALUT as illustrated in Figure L7.2.

Compile your project and download it to the Digilab-IIE board. Verify it’s functionality by using MATLAB (and the MATLAB-Xilinx interface
**Figure L7.2:** This is the overall structure of the Serial Distributed Arithmetic implementation of an IIR filter. Note that the address input to the DALUT is formed from delayed samples of the input signal and delayed samples of the output signal.

introduced in Laboratory 4) to generate ideal and experimental frequency response magnitude plots and overlay them on the same figure.

**Task 4: Optional—Speed/Area Optimizations for DA**

Choose one of the following:

1. Using one of the methods discussed in class, modify the IIR filter from Task 3 to use less than half the area of the SDA implementation. (Note: this doesn’t include the space used by the UART module. You may have to compile the filter separately to get an accurate measurement of the filter’s size.) Verify that it’s frequency response is identical to that of the SDA implementation.

2. Using one of the methods discussed in class, modify the IIR filter from Task 3 to operate in half the time of the SDA implementation. Verify that it’s frequency response is identical to that of the SDA implementation.
Report for Laboratory 7

The report for Laboratory 7 is due at the beginning of the next lab period (Nov. 4). For this report, turn in the following items:

1. Completed Check-off Sheet.
2. Properly documented VHDL code for Task 1.
3. Properly documented MATLAB plots of the frequency response from Task 1.
4. The utilization information for Task 1.
5. IIR filter parameters generated from Task 2.
7. Properly documented MATLAB plots of the frequency response from Task 3.
8. The utilization information for Task 3.